



## DE 196 15 177 C 2

## Beschreibung

Die vorliegende Erfindung bezieht sich auf ein Verfahren zum Ändern einer Funktion eines Druckertreibers, um eine ordnungsgemäße Handhabung von Spezialbedarfsanwendungen zu ermöglichen, die entdeckt werden, nachdem der Druckertreiber freigegeben ist.

Die am verbreitetsten verwendeten Betriebssysteme bei Tischcomputern sind DOS (Disk Operating System) und Windows, die beide Produkte der Microsoft Corporation, Redmond, Washington, sind. Das DOS-Betriebssystem erfordert, daß Anwendungen ein eingebettetes Druckertreibermodul aufweisen, das Kommunikationen mit einem verbundenen Drucker ermöglicht. Eine Anwendung muß daher einen in dieselbe eingebetteten geeigneten Druckertreiber für einen angeschlossenen Drucker aufweisen. Wenn zwei oder mehr Drucker unterschiedlicher Arten mit einem Computer verbunden sind, der DOS-Anwendungen aufweist, muß jede Anwendung, die die Verwendung der Drucker erfordert, eingebettete Druckertreiber für jeden Druckertyp aufweisen. Im Gegensatz dazu verwendet das Windows-Betriebssystem getrennte Druckertreibermodule, wobei ermöglicht ist, daß alle Anwendungen, die für ein Windows-Betriebssystem geschrieben sind, ein Druckertreibermodul auf eine generische Art und Weise verwenden. Folglich weisen Windows-Anwendungen eine Standardschnittstelle auf, die an das bereits existierende Druckertreibermodul anpaßt. Wenn es erforderlich ist, daß ein Computer schnittstellenmäßig mit zwei oder mehr unterschiedlichen Typen von Druckern verbunden ist, erfordert jeder einen getrennten Druckertreiber, wobei das Windows-Betriebssystem ein Druckertreibermodul für jeden der Drucker eines unterschiedlichen Typs erfordert.

Gelegentlich kann ein Konflikt zwischen einem Druckertreibermodul und einer Windows-Anwendung eine unerwartete Ausgabe auf dem Drucker zur Folge haben. Das Druckertreibermodul kann konfiguriert sein, um spezielle Dinge für die Spezialbedarfsanwendungen durchzuführen, die vor der Freigabe des Druckertreibermoduls bekannt sind. Die meisten Druckertreiber hart-codieren diese Aktionen in den Treiber selbst, wobei dann, wenn eine Spezialbedarfsanwendung ein Drucken beginnt, der Treiber seine Funktionalität in einigen Bereichen ändert, um die Anforderungen dieser Anwendung aufzunehmen.

Die Hardcodierung dieser Spezialfälle erfordert eine Vorkenntnis der einzelnen Spezialfälle vor der Freigabe des Druckertreibermoduls für eine allgemeine Verwendung. Jedoch wird gelegentlich eine Spezialbedarfsanwendung am Einsatzort identifiziert, nachdem ein Treiber für eine allgemeine Verwendung freigegeben wurde. Vor der vorliegenden Erfindung gab es keine Möglichkeit, außer dem Freigeben eines neuen Druckertreibermoduls, diese neu entdeckten Spezialbedarfsanwendungen am Einsatzort zu korrigieren.

In dem Buch von Alan Simpson: "Mastering Word Perfect 5.1 & 5.2 for Windows", 1993, Seiten 296 bis 297 und 1126 wird beschrieben, auf welche Weise zwischen einer Mehrzahl von Druckern ausgewählt werden kann, wobei jedoch nur ein Druckertreiber ausgewählt wird. Im Anhang D wird beschrieben, wie ein Merkmal eines Druckertreibers geändert wird, wobei diese Änderung keiner aufrufenden Anwendung zugeordnet ist.

Die US-A-5.228,118 betrifft ein Drucksystem, bei dem ein Druckertreiber auf der Grundlage von Übersetzern (Interpreters), welche auf dem Drucker verfügbar sind, ausgewählt wird. Das System umfaßt einen Übersetzer, um Druckdaten zu übersetzen, und einen Datenprozessor, der mit dem Drucker verbunden ist. Der Datenprozessor umfaßt eine Vielzahl von Druckertreibern, über welche die Druckdaten an den Drucker gesendet werden. Der Datenprozessor wählt einen der Druckertreiber, der mit dem Übersetzer kompatibel ist, auf der Grundlage von Übersetzer-Identifikationsdaten aus. Die Funktion eines Druckertreibers wird jedoch nicht verändert.

Die JP-A-2226327 betrifft einen Drucker, welcher eine Einrichtung aufweist, die es dem Druckertreiber oder dem Anwendungsprogramm ermöglicht, den Drucker auf schnelle Art und Weise in einen vorbestimmten Druckmodus durch Absenden eines geeigneten Steuerungscodes zu konfigurieren. Eine Änderung der Funktion des Druckertreibers selbst erfolgt nicht, sondern nur eine Änderung der Funktion des Druckers auf der Grundlage eines Steuerungscodes, welcher von dem Computer empfangen wurde.

Die Aufgabe der vorliegenden Erfindung besteht darin, ein einfaches und bequemes Verfahren zum Ändern der Funktion eines Druckertreibers zu schaffen, um bestimmten Anwendungsprogrammen für bestimmte Drucker voreingestellte, vom Standarddruckertreiber abweichende Funktionen zuzuordnen.

Diese Aufgabe wird durch ein Verfahren gemäß dem einzigen Patentanspruch gelöst.

Wenn die Anwendung Dienste der Druckertreiber anfordert, werden alle Datenstrukturen abgetastet, wobei nach einem Eintrag für die anfordernde Anwendung gesucht wird. Wenn ein Eintrag gefunden wird, wird die Funktions-Flag verwendet, um eine Änderung der Funktion des Druckertreibers zu leiten. Wenn kein Eintrag gefunden wird, wird eine permanente Liste nach einem Eintrag für die Anwendung abgesucht. Wenn ein Eintrag in der permanenten Liste gefunden wird, wird derselbe wiederum verwendet, um eine Änderung der Funktion des Druckertreibers zu leiten.

Die vorliegende Erfindung arbeitet in einem System, das einen Host-Computer und einen Drucker, der mit dem Host-Computer verbunden ist, aufweist. Eine Anwendung ermöglicht dem Host-Computer, ein Programm durchzuführen. Anwendungen kommunizieren durch einen Druckertreiber mit dem Drucker. Es existiert ein Dienstprogramm, um dem Host-Computer anzuzeigen, daß die Anwendung eine spezielle Handhabung durch den Druckertreiber erfordert. Der Host-Computer weist eine Permanentspeichervorrichtung auf, um die Anzeige zu speichern, wenn der Host-Computer abgeschaltet wird. Ein Lokalspeicher wird durch den Host-Computer verwendet, um Informationen zu speichern, während der Host-Computer eingeschaltet ist.

Wenn die Anwendung Dienste des Druckertreibers anfordert, wird der Lokalspeicher abgetastet, wobei nach einem Eintrag für die vorliegende Anwendung gesucht wird. Wenn bei dem Abtasten ein Eintrag gefunden wird, wird eine Funktion des Druckertreibers geändert, um diese Anwendung aufzunehmen.

Bevorzugte Ausführungsbeispiele der vorliegenden Erfindung werden nachfolgend bezugnehmend auf die beiliegenden Zeichnungen näher erläutert. Es zeigen:

Fig. 1 ein Blockdiagramm, das innere Aspekte eines Host-Computers zeigt;

Fig. 2 die Benutzeroberfläche des Dienstprogramms App-Flag; und

Fig. 3 ein logisches Flußdiagramm der Funktion SetupApp-Flags 0.

## DE 196 15 177 C 2

Die vorliegende Erfindung ist nicht auf das spezielle Ausführungsbeispiel, das hierin veranschaulicht ist, begrenzt. Wie speziell in Fig. 1 gezeigt ist, ist ein Host-Computer 10 über eine Standard-I/O-Schnittstelle 14 (I/O = Input/Output = Eingabe/Ausgabe) mit einem Drucker 12 verbunden. Zu Zwecken dieser Beschreibung sei angenommen, daß der Host-Computer 10 ein Personalcomputer ist. Ein Verstehen der inneren Abläufe des Druckers 12 ist nicht notwendig, um die vorliegende Erfindung zu verstehen.

Der Host-Computer 10 weist eine Zentralverarbeitungseinheit 16 und einen Direktzugriffsspeicher (RAM; RAM = Random Access Memory) auf, der in eine Anzahl von Abschnitten segmentiert ist. Ein RAM-Abschnitt 18 enthält einen Software-Code zum Steuern der Anwendung 15 des Host-Computers, von Druckertreiberfunktionen 17 und einer Benutzerschnittstelle 13. Der RAM-Abschnitt 18 enthält ferner Systemvariablen und das Betriebssystem des Host. Ein Non-Volatile-Speicher 20 (ROM; ROM = Read Only Memory) weist eine Firmware zum Steuern des elementaren Eingabe/Ausgabe-Systems (BIOS; BIOS = Basic Input/Output-System) und einen Code zum Steuern weiterer Host-Funktionen auf. Ein RAM-Abschnitt 22 enthält eine Druckertreibersoftware, um zu ermöglichen, daß der Host-Computer 10 entweder in einem Sless-Modus 23 (Sless = glatt) oder einem PCL-Modus 24 arbeitet. In diesem Fall bedeutet PCL "Printer Control Language" (= Druckersteuersprache), was ein standardmäßiger, weit verbreiteter Treiber für Drucker ist. Fachleute sollten offensichtlich sein, daß die vorliegende Erfindung nicht auf eine spezielle Sprache oder eine Anzahl von Sprachen begrenzt ist. Ein weiterer Abschnitt 25 des RAM ist reserviert, um als ein Puffer zu wirken, um Rasterbilddaten zu enthalten, die durch den Sless-Treiber 23 formatiert wurden und bereit sind, um über die I/O-Hardware und das Treibermodul 26 zu dem Laserdrucker 12 übertragen zu werden. In Fig. 1 ist ferner ein Speichermedium 21 gezeigt, das ein Festplattenlaufwerk und ein Wechseldiskettenlaufwerk aufweist.

Wie speziell in Fig. 2 gezeigt ist, liefert das bevorzugte Ausführungsbeispiel der vorliegenden Erfindung die Fähigkeit, eine Anwendung 15 am Einsatzort als "Spezialfall" zu behandeln. Diese spezielle Behandlung wird durch eine Anwendungsbasis unter Verwendung eines Passwort-geschützten Dienstprogramms auf eine Anwendung angewendet. Bei dem bevorzugten Ausführungsbeispiel ist die Liste von Anwendungs-Flags, die angewendet werden können, ziemlich gering, deckt jedoch einen großen Bereich der Druckprobleme ab, welche alle Spezialfallanwendungen einschließen, die vor der Freigabe des Druckertreibermoduls identifiziert wurden.

Ein Passwort ist erforderlich, um das Dienstprogramm zu starten, wodurch die Wahrscheinlichkeit einer unbeabsichtigten Verwendung reduziert wird. Der Benutzer am Einsatzort führt das Dienstprogramm auf die normale Art und Weise aus, wobei er das Passwort eingibt, wenn es angefordert wird. Sobald das Dienstprogramm gestartet ist, wird dem Benutzer eine Anwendungs-Flag-Benutzerschnittstelle 13 geboten, wie in Fig. 2 gezeigt ist. Um eine Anwendung 15 als einen Spezialfall zu behandeln, drückt der Benutzer den Knopf "NEU", was die Standarddatei-Offen-Dialogbox in den Vordergrund bringt. Als nächstes lokalisiert der Benutzer die Anwendungsdatei, die die Probleme bewirkt, markiert dieselbe und drückt OK. Das Dienstprogramm "AppFlag" verifiziert, daß diese ein ausführbares Programm ist, lokalisiert den programminternen Modulnamen, lokalisiert die Versions-Betriebsmittelinformationen in der Datei, wenn solche vorliegen, und zeigt den Modulnamen und die Versionsinformationen in der Listbox an. Ein Eintrag wird in der AppFlag-Datenbank 21 gemacht, der den Modulnamen, die Versionsnummer, eine Nummer, die das momentane Betriebssystem darstellt, und einen AppFlag-Eintrag von Null enthält. Der Null-Eintrag zeigt an, daß für diese Anwendung gegenwärtig keine AppFlags gesetzt sind. Um eine AppFlag für eine Anwendung 15 zu setzen, wählt der Benutzer dieselbe zuerst in der Listbox aus und klickt die geeignete Kontrollbox an. Alle Änderungen, die durchgeführt werden, werden nicht in der Datenbank 21 gespeichert, bis der Anwenden-Knopf gedrückt wird.

Nachfolgend wird detaillierter auf Fig. 2 Bezug genommen, in der die AppFlag-Benutzerschnittstelle gezeigt ist. Bei dem bevorzugten Ausführungsbeispiel kann der Benutzer aus 10 Flags auswählen. Für Fachleute ist es offensichtlich, daß diese 10 Flags nur darstellend für die üblichsten Probleme sind, die gegenwärtig identifiziert sind. Die Flag mit der Nummer 1, die mit "keine Abbruch-Prozedur" ("No Abort Procedure") bezeichnet ist, zeigt an, daß diese Anwendung 15 keine Abbruchprozedur verwendet. Die Flag 2, die mit "nur 300 DPI" ("300 DPI only") bezeichnet ist, zwingt den Druckertreiber 22, für diese spezifizierte Anwendung stets mit 300 dpi zu drucken. Es sei nun auf die Flag mit der Nummer 3 mit dem Titel "Technologie = Raster" verwiesen. Die Flag 3 bewirkt, daß der Druckertreiber 22 seine Drucktechnologie als Raster zu der spezifizierten Anwendung 15 berichtet. Die Flag 4 ist mit "keine Font-Maschinen-Drehung" ("No Font Engine Rotation") bezeichnet. Das Setzen dieser Flag setzt den Druckertreiber 22 außerstand, eine Textdrehung durchzuführen, wobei die Anwendung 15, oder höchstwahrscheinlich der Videotreiber gezwungen wird, die nächste Drehung für den Druckertreiber 22 durchzuführen. Die Flag 5, "unterstützte Bandinfo" ("support bandinfo"), bewirkt, daß der Druckertreiber 22 den "veralteten" Bandinfo-Druckervodewechsel unterstützt. Die Flag 6, "ignorierte Modelländerung" ("ignore model change"), zeigt an, daß einige Anwendungen die PDEVICE-Struktur (eine Datenstruktur, die durch den Druckertreiber verwendet und durch GDI beibehalten wird) falsch handhaben, was zur Folge hat, daß ein Teil derselben weggeworfen wird. Dieser Datenverlust bewirkt, daß der Druckertreiber 22 glaubt, daß die Konfiguration falsch ist, und derselbe den Benutzer informiert, daß die Drucksoftware neu installiert werden muß. Wenn diese Flag gesetzt ist, ignoriert der Druckertreiber 22 diese Bedingung und verwendet voreingestellte Werte für die fehlenden Daten. Die Flag 7, "keine Skalierungsunterstützung" ("no scaling support") zwingt den Druckertreiber 22, der Anwendung 15 zu berichten, daß er nicht in der Lage ist, eine Skalierung durchzuführen. Die Flag 8, "verwende logischen Tomamen" ("use logical port name") zeigt an, daß die Anwendung 15 den logischen Tomamen im Gegensatz zu dem unmittelbaren Tomamen verwendet wird. Die Flag 9, "kein manueller Vorschub zuerst" ("no manual feed first"), zwingt den Druckertreiber 22, sicherzustellen, daß "manueller Vorschub" niemals der erste Behältername (bin name) ist, der in einer Anwendung 15 berichtet wird. Schließlich korrigiert die Flag 10, "wechsle Papierabmessungen" ("swap paper dimensions") ein Mißverständnis zwischen dem Druckertreiber 22 und der Anwendung 15 bezüglich den Papierabmessungen.

Der Druckertreiber, wie er vorher beschrieben wurde, weist eine hart-codierte Liste derjenigen bekannten Anwendungen auf, die eine spezielle Handhabung benötigen. Bei dem bevorzugten Ausführungsbeispiel ist diese Liste in einem Array von Datenstrukturen gespeichert, die wie folgt definiert ist:

## DE 196 15 177 C 2

```

typedef struct {
    LPSTR    lpModuleName;
    5      DWORD    dwFileVersionMS;
            DWORD    dwFileVersionLS;
            DWORD    dwOSValid;
    10      DWORD    dwAppFlags;
} APPFLAG_INFO, For *LPAPPFLAG_INFO.

Ein darstellendes Muster der hart-verdrahteten Liste ist nachfolgend gezeigt:
15 {  "EXCEL",

            APPFLAG_ALL_VERSIONS,
            APPFLAG_ALL_VERSIONS,
    20      APPFLAG_OS_ALL, APPFLAG_NODMSCALESUPPORT |
APPFLAG_NOROTENGREALIZE },

    25 {  "Calendar",

            APPFLAG_ALL_VERSIONS,
            APPFLAG_ALL_VERSIONS,
    30      APPFLAG_OS_ALL,
            APPFLAG_MAXRES300 },

    35 {  "CARDFILE",

            APPFLAG_ALL_VERSIONS,
            APPFLAG_ALL_VERSIONS,
    40      APPFLAG_OS_ALL,
            APPFLAG_MAXRES300 },

    45 {  "MSVC",

            APPFLAG_ALL_VERSIONS,
    50      APPFLAG_ALL_VERSIONS,
            APPFLAG_OS_W3X,
            APPFLAG_NOABORTPROC },

```

55 Bei dem bevorzugten Ausführungsbeispiel weist der Druckertreiber 22 einen Zugriff auf eine externe Datenbank 21 zusätzlich zu dem auf die oben gezeigte hart-codierte Liste auf. Die externe Datenbank 21 wird durch das Dienstprogramm AppFlag beibehalten, wie oben beschrieben ist.

60 Wenn eine Anwendung 15 irgendeinen Dienst des Druckertreibers 22, 17 anfordert, bestimmt der Druckertreiber 22, ob für die Anwendung 15 irgendwelche speziellen Bedürfnisse identifiziert wurden, indem beispielsweise unter Verwendung der folgenden Codezeile ein Aufruf an die SetupAppFlags()-Funktion erteilt wird:

```
lpdv → dwAppFlags = SetupAppFlags();
```

65 Diese Funktion gibt einen Wert zurück, der eine Bit-abgebildete Beschreibung aller speziellen Bedürfnisse (wenn solche existieren) der Anwendung 15, die die Dienste des Druckertreibers 22 anforderte, enthält. Das Verhalten des Druckertreibers 22 wird modifiziert, wenn ein spezifisches Bit in dem Ergebnis gesetzt ist.

Wie in Fig. 3 gezeigt ist, sammelt die Funktion SetupAppFlags() zuerst Informationen 201 über die Anwendung 15,

## DE 196 15 177 C 2

die die Dienste des Druckertreibers 22 anfordert. Die gesammelten Informationen enthalten den internen Modulnamen, den vollen Pfadnamen der ausführbaren Datei und Versionsnummern der Anwendung. Als nächstes wird die AppFlag-Datenbank abgesucht 202, um zu bestimmen, ob durch das AppFlag-Dienstprogramm irgendwelche Spezialbedarfsanwendungen eingestellt wurden. Eine Übereinstimmung 203 wird durch das Vergleichen der Modulnamen, der Versionsnummern und des Betriebssystems, die in der AppFlag spezifiziert sind, bestimmt. Wenn eine Übereinstimmung in der externen AppFlag-Datenbank 21 gefunden wird, werden die AppFlags für die Anwendung 15 zurückgegeben 206.

Wenn keine Übereinstimmung in der externen AppFlag-Datenbank 21 gefunden wird, wird die AppFlag-Hartcode-Tabelle abgesucht 204. Flags für jede übereinstimmende Anwendung 15 (es können für eine spezifische Anwendungs- und Versionsnummern-Kombination mehrere existieren) werden Bit-weise miteinander ODER-verknüpft, um das Ergebnis 206 zu bilden.

Indem die externe Datenbank 21 zuerst durchsucht wird, und die Suche dann angehalten wird, wenn eine Übereinstimmung gefunden ist, können die hart-codierte AppFlags in dem Feld überschrieben werden, ohne den Code des Druckertreibers 22 zu modifizieren.

Alle Zugriffe auf die externe Datenbank 21 erfolgen durch eine Instanziierung (instantiation) (ein Objekt) der DCustAppFlags-Klasse. Die Deklaration der DCustAppFlags-Klasse ist nachfolgend gezeigt:

```
typedef struct APPINFOARR {
    char    szModule(10);
    APPFLAG_INFO ai;
} APPINFOARR, FAR *LPAPPINFOARR;

class DCustAppFlags {
public:    // Construction / Destruction
    DCustAppFlags();
    ~DCustAppFlags();
public:    // Public data.
public:    // Operation.
    int Refresh();
    in GetCount() { return m_nElems; }
    LPAPPFLAG_INFO operator [] ( int index );
    LPAPPFLAG_INFO GetAIPtr( LPSTR lpszModule );
protected: // Hidden data.
    int m_nElems;
    LPAPPINFOARR m_nlpArr;
protected: // Hidden operations.
    int Load();
    void Unload();
protected: // Protected helper operations.
    LPSTR AllocStrPtr ( DWORD dwSize );
    LPAPPINFOARR AllocAppInfoArray ( int num );
    BOOL GetAppFlagIniEntry (LPCSTR lpszItem,
APPFLAG_INFO & APPInfo);
    BOOL IsHexValue ( char c);
    int HexValue( char c );
    DWORD str2dw( char *str );
    int GetItemNames(LPSTR lpszBuf, int nBufSize
); };
```

## DE 196 15 177 C 2

Der Druckertreiber 22 bildet ein Objekt dieser Klasse mit einer Aussage wie der folgenden als eine Instanz;

DCustAppFlags custAppFlags.

- 5 Ein Objekt dieser Klasse läßt bei der Instanzbildung alle AppFlags aus der externen AppFlag-Datenbank 21 in den Speicher 18. Dies liefert eine schnelle Antwort, wenn der Druckertreiber 22 eine Suche durchführt. Eine Suche wird durchgeführt, immer wenn ein Aufruf der SetupAppFlag()-Funktion durchgeführt wird, wie oben beschrieben ist. Es wurde erwähnt, daß die externe AppFlag-Datenbank 21 zuerst überprüft wird. Dies geschieht in SetupAppFlag(), indem die GetAppFlag()-Mitgliedfunktion des custAppFlags-Objekts aufgerufen wird. Diese Mitgliedfunktion durchsucht ihre interne AppFlag-Datenbank und gibt einen Zeiger zu der APPFLAG INFO-Struktur zurück, wenn eine Übereinstimmung gefunden wird.

## Patentansprüche

- 15 Verfahren zum Ändern einer Funktion eines Druckertreibers (22) mit folgenden Schritten:  
 Aktivieren eines Dienstprogramms (Fig. 2);  
 Auswählen einer Anwendung (Fig. 2) über das Dienstprogramm;  
 Anzeigen der Funktion durch Setzen einer Funktions-Flag (Fig. 2);  
 Zuordnen der Funktions-Flag zu der Anwendung in einer Datenstruktur;  
 20 Speichern der Datenstruktur in einer Speichereinrichtung (21, 18);  
 Erfassen (201), wenn die Anwendung Dienste des Druckertreibers (22) anfordert;  
 Lesen der Datenstruktur aus der Speichereinrichtung (21, 18);  
 erstes Durchsuchen (202) der Datenstruktur nach der Funktions-Flag, die der Anwendung zugeordnet ist;  
 wenn der Schritt des ersten Durchsuchens (202) die Funktions-Flag, die der Anwendung zugeordnet ist, findet,  
 25 Wiedergewinnen (206) der Funktions-Flag;  
 Verwenden der Funktions-Flag, um eine Änderung der Funktion des Druckertreibers (22) durchzuführen;  
 wenn der Schritt des ersten Durchsuchens (202) die Funktions-Flag, die der Anwendung zugeordnet ist, nicht findet,  
 zweites Durchsuchen (204) einer Festtabelle nach einem Eintrag, der der Anwendung zugeordnet ist; und  
 wenn der Schritt des zweiten Durchsuchens (204) den Eintrag findet, Ermöglichen (206), daß der Eintrag eine Än-  
 30 derung der Funktion des Druckertreibers (22) durchführt.

Hierzu 3 Seite(n) Zeichnungen

35

40

45

50

55

60

65

JAN-12-2004 09:44

BOISE LEGAL

208 396 3958 P.10/13

- Leerseite -

ZEICHNUNGEN SEITE 1

Nummer:

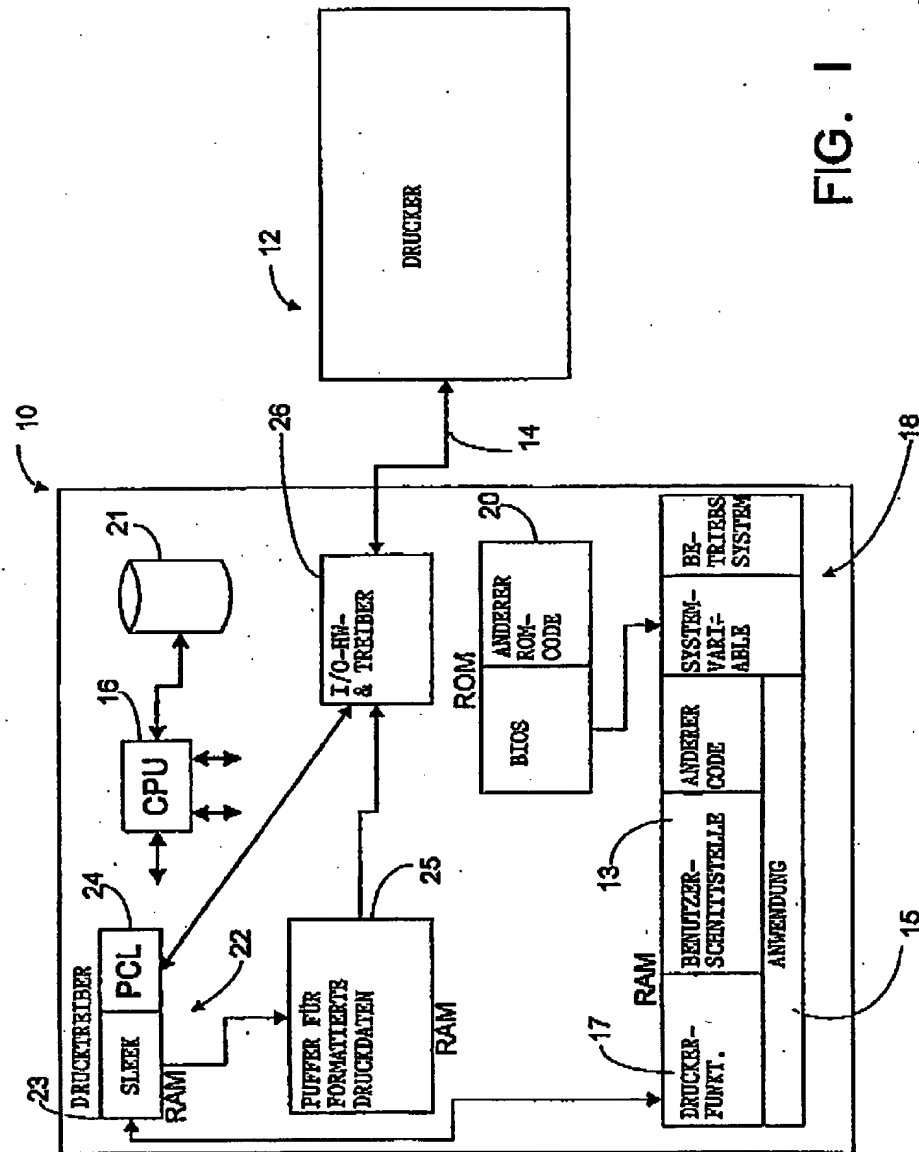
DE 196 15 177 C2

Int. Cl.:

G 08 F 3/12

Veröffentlichungstag:

11. März 1999



802 170/146

ZEICHNUNGEN SEITE 2

Nummer:

DE 198 15 177 C2

Int. Cl.:

G 06 F 3/12

Veröffentlichungstag:

11. März 1999

**SETZE ANWENDUNGS-FLAGS**

**ANWENDUNGEN**

1. ☐ KEINE ABRUCH-PROZEDUR

2. ☐ NUR 300 DPI

3. ☐ TECHNOLOGIE = RASTER

4. ☐ KEINE FONT-MASCHINEN-DREHUNG

5. ☐ UNTERSTÜTZE BANDINFO

6. ☐ IGNORIERE MODELLÄNDERUNG

7. ☐ KEINE SKALIERUNGSUNTERSTÜTZUNG

8. ☐ VERWENDE LOGISCHEN TORNAMEN

9. ☐ KEIN MANUELLER VORSCHUB ZUERST

10. ☐ WECHSELE PAPIERABMESSUNG

**SCHLIESSE**

**NEU**

**WENDE AN**

**LÖSCHE**

**ÜBER**

FIG. 2

802 170/146

ZEICHNUNGEN SEITE 3

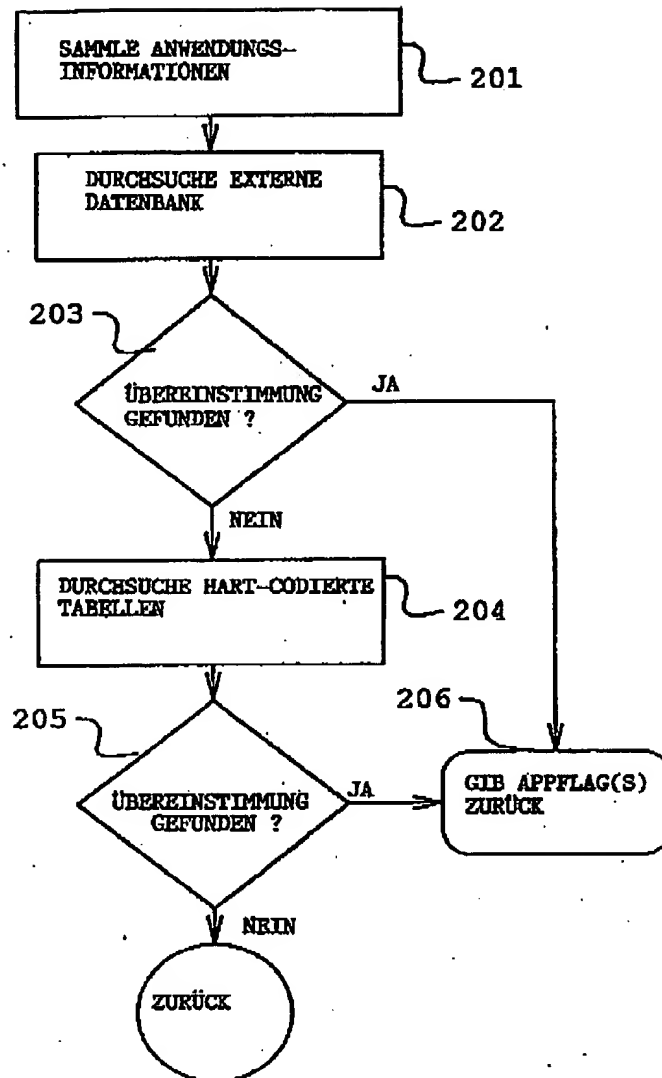
Nummer:  
Int. Cl. 5:  
Veröffentlichungstag:DE 196 15 177 C2  
G 06 F 3/12  
11. März 1999

FIG.3

802 170/146

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**